

# Planificación para interceptación de objetivos: Integración del Método Fast Marching y Risk-RRT

David Garzón-Ramos, Mario Garzón y Antonio Barrientos  
Centro De Automática y Robótica UPM-CSIC  
Calle José Gutiérrez Abascal, 2. 28006 Madrid, España  
dgarzon@etsii.upm.es, ma.garzon@upm.es, antonio.barrientos@upm.es

## Resumen

*Este artículo presenta un nuevo enfoque para la interceptación de objetivos móviles en escenarios complejos y de gran tamaño. El planificador para interceptación se basa en el algoritmo Risk-RRT. Para mejorar el desempeño del algoritmo original se ha realizado la adaptación e integración de técnicas basadas en el Método de Fast Marching. Las modificaciones realizadas han permitido integrar de forma efectiva la información del entorno dentro del planificador y mejorar sus capacidades. La integración completa es capaz de planificar on-line y seguir la trayectoria, al mismo tiempo que ejecuta las funciones de interceptación y evasión de obstáculos.*

**Palabras clave:** Fast Marching Method, Planificación para Interceptación, Risk-RRT.

## 1 Introducción

Mejorar las capacidades de sistemas para vigilancia autónoma de infraestructuras es una tarea sobre la que se ha puesto gran atención en los últimos años. En este ámbito, el desarrollo de metodologías para el seguimiento e interceptación de objetivos móviles empleando vehículos terrestres no tripulados (UGV) es una tarea que se muestra prometedora en aplicaciones de seguridad y defensa. Existen diversos retos asociados al diseño e implementación de estas funcionalidades en sistemas robotizados, sin embargo, es posible destacar cuatro líneas principales de interés: detección y localización del objetivo, identificación y predicción de su trayectoria, planificación de una ruta para interceptarlo y finalmente, la ejecución de la trayectoria. El trabajo presentado en este artículo se enfoca en mejorar las etapas de predicción y planificación y se espera que pueda ser empleado en tareas de vigilancia de infraestructuras críticas de gran escala.

El objetivo principal de este trabajo es realizar una mejora de algoritmo planificación para interceptación desarrollado por Garzón *et. al.* [1] mediante la integración de mapas de distancia

creados utilizando el Método de Fast Marching (FMM) [10], en un planificador basado en Rapidly-exploring Random Trees (RRT) [5]. El sistema original propuso un algoritmo para la interceptación de objetivos utilizando UGV's, con capacidad de evasión de obstáculos estáticos y dinámicos. Entre las ventajas de este algoritmo se pueden nombrar su completo acople a la arquitectura de navegación de robots móviles terrestres y que permite la incorporación de nuevas fuentes de información gracias a su diseño modular. El desarrollo presentado en este trabajo mantiene dichas características a la vez que mejora su rendimiento y adaptabilidad al agregarle una nueva capa de información.

Otros trabajos anteriores han buscado optimizar las capacidades de algoritmos basados en RRT mediante el Método de Fast Marching (FMM). Estos trabajos enfocan su desarrollo en los llamados Fast Marching Trees [2, 3]. Este tipo de algoritmo difiere de la implementación convencional de RRT debido a que emplea las capacidades del FMM para calcular, de forma recursiva, la distancia de recorrido entre nodos, en lugar de la distancia euclidiana normalmente utilizada. Así, busca generar trayectorias más cercanas al desplazamiento real del robot. El gran problema de esta implementación es que ejecuta de forma exhaustiva el FMM sobre cada nodo y esto tiene un coste computacional muy elevado.

Este artículo es el resultado de primeros estudios para optimizar el funcionamiento del algoritmo Risk-RRT en tareas de interceptación de intrusos, orientado a su aplicación en la protección de infraestructuras críticas. LA principal novedad del trabajo aquí presentado es la integración del FMM, que en este caso se hace mediante la inclusión global la información del mapa sobre todo el árbol, en lugar de realizarlo de forma local en cada nodo. La ejecución del FMM se realiza desde la posición del objetivo a interceptar para obtener la distancia desde ese punto a cualquier lugar en el escenario. De esta forma, dicha información puede ser aprovechada para estimar de forma correcta la distancia de cualquier nodo del árbol hasta la posición del objetivo y adicionalmente para pre-

decir las posibles rutas que este va a emplear para desplazarse.

El trabajo se estructura de la siguiente forma: en la sección 2 se hace una descripción del algoritmo de interceptación basado en Risk-RRT. En la sección 3 se detalla la implementación del Método de Fast Marching. La sección 4 resume la obtención del mapa para la predicción de rutas. En la sección 5 se describe la integración de los algoritmos. La sección 6 muestra los experimentos y resultados y finalmente en la sección 7 se dan las conclusiones.

## 2 Interceptación de objetivos móviles mediante Risk-RRT

Risk-RRT es una estrategia de planificación propuesta en [4], la cual ha sido diseñada para operar en escenarios dinámicos y en presencia de incertidumbre. En esta, el riesgo de colisión es calculado usando modelos probabilísticos que representan objetos estáticos y dinámicos. El proceso usado para buscar las trayectorias óptimas se basa en una extensión del algoritmo Rapidly-exploring Random Tree (RRT) [5].

En el algoritmo Risk-RRT, el riesgo de colisión puede ser definido como la probabilidad de que el robot choque con algún obstáculo si desde el estado actual es aplicada una acción de control determinada. Con un entorno previamente conocido y representado por celdas de ocupación, la probabilidad de colisión con obstáculos estáticos está definida por la cercanía de los estados alcanzables por el robot a una celda marcada como ocupada. Por este motivo, la probabilidad de colisión en dichas celdas no cambia en el tiempo. En cambio, en presencia de obstáculos dinámicos, la probabilidad de colisión se extiende desde la posición instantánea del obstáculo en  $t_0$  hasta la posible posición que ocupará en  $t_n$ .

En trabajos anteriores se han realizado modificaciones de este algoritmo que aprovechan la estimación del riesgo de colisión con un objeto móvil con el fin de determinar la ruta óptima para interceptarlo. Siendo esta una implementación del algoritmo RRT, la ruta es extraída de un árbol de configuraciones alcanzables por el robot. Los nodos del árbol, o puntos de paso del robot, son seleccionados de acuerdo peso ponderado entre la probabilidad de colisión con el objetivo móvil y la longitud estimada de la trayectoria. Una completa descripción de este trabajo puede ser encontrada en [1].

A pesar de los prometedores resultados de la técnica, existen inconvenientes asociados a la falta de integración del algoritmo con el entorno en

donde se desarrolla. El primer problema se presenta en la estimación de las distancias, en donde se emplea la distancia euclídea para calcular la distancia más corta entre un nodo y el objetivo móvil. El segundo problema está asociado al uso de modelos probabilísticos para la predicción del desplazamiento del objetivo móvil que no toma en cuenta su localización en el espacio. En ambos casos no se toma en cuenta la presencia de obstáculos en el entorno, lo que fácilmente se traduce en la aparición de soluciones con mínimos locales en la planificación y estimaciones de la probabilidad que no pueden ser extendidas a largos periodos de tiempo.

A las secciones posteriores se expone la implementación del Método Fast Marching y una primera aproximación para solucionar estas problemáticas mediante la combinación del algoritmo Risk-RRT con el Método Fast Marching.

## 3 Método Fast Marching

El Método de Fast Marching (FMM) es una técnica para seguir la evolución de un frente de onda que se expande en un medio viscoso. En este sentido, el frente de onda será una curva que separa el medio en una región interior y exterior, cuya evolución en el tiempo corresponde a un desplazamiento en la dirección normal al frente de onda. En el algoritmo original, el medio se considera isotrópico y la velocidad de expansión está determinada por la magnitud de la velocidad de desplazamiento en el medio, la cual no necesariamente es constante pero siempre debe ser positiva. Las variaciones del algoritmo original[6] modelan la velocidad de traslado con magnitud y dirección, considerando así modelos anisotrópicos del espacio, en donde el frente de onda evoluciona de distintas formas según sea su dirección de movimiento. Una velocidad de desplazamiento de cero, indica la imposibilidad del movimiento de la onda por dicho punto, tanto en el modelo isotrópico como en el anisotrópico.

En un punto dado, el movimiento del frente de onda esta descrito por la ecuación Eikonal:

$$\|\nabla T(x)\| F(x) = 1 \quad (1)$$

en donde  $T$  es el tiempo de llegada del frente en un punto  $x$  y  $F \geq 0$  es la velocidad del frente en el punto  $x$ . Dado que el frente de onda únicamente se puede expandir, existe un valor único para  $T$ . En los siguientes apartados se describen los pasos previos a la construcción del mapa FMM, así como el algoritmo empleado para realizar la expansión de onda.

### 3.1 Mapa de Costos

El primer paso para realizar la implementación del algoritmo FMM es obtener una representación del espacio que se ajuste al model de interceptación de objetivos. Con este fin se construye un mapa de costos el cual corresponde a una representación modificada del escenario original y que además incluye información sobre la dificultad de desplazarse sobre un espacio determinado en el mapa.

Primero, se emplea una transformación de la distancia para añadir costos fijos sobre las trayectorias que pueden ser seguidas por los objetivos a interceptar o los robots. En el caso de los objetivos, esta técnica es empleada para modelar el comportamientos de evasión de obstáculos y con preferencia a permanecen distanciados de las paredes. De igual forma, los robots emplean esta información para establecer rutas seguras de desplazamiento, evitando riesgos de colisión. La transformación de la distancia se traduce en un gradiente en la escala de grises que rodea los obstáculos y paredes, con valor máximo alcanzado en las cercanías de los objetos y con un decrecimiento constante en la dirección normal del objeto, hasta alcanzar una extensión de seguridad mínima que se define de forma predeterminada. De este modo, en todos los procesos asociados al FMM se toma en cuenta la planificación de rutas con evasión de obstáculos tanto de los robots como de los objetivos.

Dado que la función del mapa de costos es brindar una representación del mapa que se ajusta a la tarea en desarrollo, en este caso también se contempla una etapa de escalado sobre el tamaño del mapa original. Este cambio en el mapa permite añadir o extraer pixeles de la imagen con el fin de optimizar ciertos procesos, sin que sea alterada en gran medida su composición general.

El proceso de escalado se realiza mediante un algoritmo de interpolación bilineal, que asegura que la posición en el mapa resultante tenga una correcta traducción que el mapa original. Cuando la imagen es escalada, a cada pixel original se le asigna uno o varios pixeles en la gráfica nueva, según el factor de escala deseado. Este método permite determinar el valor de intensidad del nuevo pixel, a través de una media ponderada de las intensidades de los pixeles conocidos en una vecindad de grados. En el Algoritmo 1 se resume el proceso de construcción del mapa de costos.

A partir del vector  $Vm\{0, 1, \dots, k\}$ , que representa las celdas de ocupación del mapa, se obtiene un nuevo vector  $Cm\{0, 1, \dots, r\}$  en donde  $r$  esta definido por el factor de escala  $C$  que relaciona la distancia en metros representada por cada pixel

---

#### Algorithm 1 Construcción del mapa de costos

---

**Require:** Vector  $Vm\{0, 1, \dots, k\}$

**Require:** Factor de escala ( $C$ ) & distancia de seguridad ( $d_s$ ).

**Ensure:** Vector  $Cm\{0, 1, \dots, r\}$ ;  $r = S_x k$

```

1: for  $i \leq k$  do
2:    $r = \text{función de escalado}(k, S)$ .
3:    $Cm(r) = \text{transformación de distancia}$ 
      $(Vm(k), d_s)$ .
4:    $r = r + 1$ .
5: end for

```

---

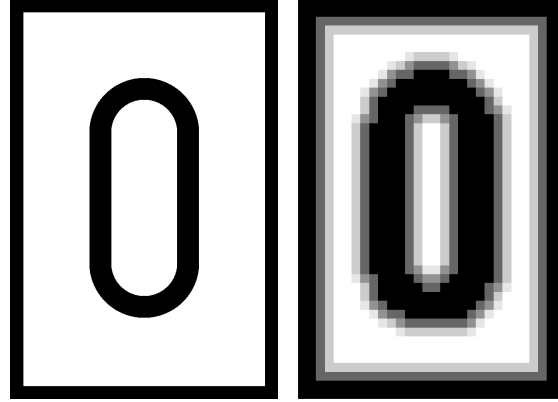


Figura 1: Variaciones de escala y distancia de seguridad en la construcción del mapa de costos.

en  $Vm$  y la distancia en metros deseada para  $Cm$ . De igual forma, se emplea la distancia de seguridad  $d_s$  para asignar un nivel de intensidad entre 0 y 100 a los pixeles que se encuentran en cercanías a obstáculos dentro del mapa.

En la Figura 1 se muestra la representación en forma de mapa de costos de un escenario arbitrario cuando es sometido al proceso de escalamiento y de transformación de la distancia. La variación de escala se ha realizado con  $0.1m$  y  $2m$  por pixel, y la distancia de seguridad con valores de 0 y  $5m$ . Se puede observar que aunque la resolución se ha reducido en más de un orden de magnitud, la imagen conserva su morfología y características más representativas. El efecto más notable de la reducción de la resolución, se evidencia en la incapacidad de generar una transformación de la distancia adecuada. Al reducir el número de pixeles por metro, disminuyen también la cantidad de intensidades de gris que pueden ser incluidas en el área de seguridad. Este efecto es poco deseado en los sistemas de navegación ya que supone un cambio brusco en la velocidad de movimiento.

### 3.2 Mapa de velocidades

A continuación el mapa de costes es ajustado al modelo de entrada del FMM. La información re-

querida para ejecutar el algoritmo es un mapa de velocidades, por este motivo el mapa de costos obtenido anteriormente es transformado al espacio de la velocidad en donde a cada pixel dentro del vector  $Cm\{0, 1, \dots, r\}$  se le asigna un nuevo valor mediante la expresión:

$$W(r) = \frac{Cm(r)}{\max(Cm)} \quad (2)$$

En donde  $W(r)$  simboliza la velocidad en el pixel  $r$  dentro del mapa de velocidades  $W$ .  $Cm(r)$  representa el costo del desplazamiento en dicho pixel y  $\max(Cm)$  corresponde al máximo valor en el mapa de costos. De este modo, la velocidad que puede ser alcanzada por la onda propagándose en un punto del mapa es inversamente proporcional al costo que supone desplazarse por ese punto. Suponiendo una velocidad máxima constante, las zonas totalmente libres de obstáculos obtendrán una velocidad máxima igual a 1 y aquellas que se encuentran ocupadas reciben una velocidad de 0 dado que es imposible desplazarse a través de ellas. En este caso, se ve reflejado el comportamiento de peatones en donde estos se mueven continuamente con velocidad constante en las áreas libres y sólo disminuyen su avance cuando se encuentran cerca de paredes u obstáculos.

### 3.3 Implementación de FMM

Dado que el algoritmo Risk-RRT se encuentra desarrollado para Robot Operating System (ROS). En este trabajo se ha realizado una adaptación para ROS de las librerías correspondientes a la Fast Marching Toolbox de Matlab®, desarrolladas por Gabriel Peyre<sup>1</sup>. En esta implementación el algoritmo de FMM se ejecuta sobre un arreglo de celdas, correspondientes a los pixeles en una imagen, y asigna a cada pixel el tiempo de llegada del frente de onda según lo expuesto en la sección 3. De forma que, para un medio con igual velocidad de propagación en el espacio, la expansión de la curva resulta en círculos concéntricos que se alejan desde el origen hasta la distancia máxima que la onda puede recorrer dentro del mapa. En el Algoritmo 2 se describen los pasos empleados en la librería para realizar la expansión de onda mediante FMM.

En el Algoritmo 2 la función distanciaEikonal esta descrita por la expresión

---

#### Algorithm 2 Construcción del mapa FMM

---

**Require:** Vector  $W\{0, 1, \dots, r\}$

**Require:** Posición inicial  $x_0$ .

**Ensure:** Vector de distancias  $D\{0, 1, \dots, r\}$ .

- 1: Distancias  $D\{0, 1, \dots, r\} = \infty$ .
  - 2: Estados  $S\{0, 1, \dots, r\} = \text{no visitados}$ .
  - 3:  $D(x_0) = 0$  &  $S(x_0) = \text{visitado}$ .
  - 4: **repeat**
  - 5:   Busca  $\min(D)$  con  $S(\min(D)) = \text{no visitado}$ .
  - 6:   Selecciona pixeles  $[n_{right}, n_{left}, n_{up}, n_{down}]$  adyacentes a  $\min(D)$ .
  - 7:    $S(\min(D)) = \text{visitado}$ .
  - 8:   **for all**  $S(n_j) = \text{no visitado}$  **do**
  - 9:      $D(n_j) = \text{distanciaEikonal desde } \min(D) \text{ a } n_j$ .
  - 10:   **end for**
  - 11: **until**  $S(0, 1, \dots, r) = \text{visitados}$
- 

$$d = \begin{cases} \frac{d_x + d_y + \sqrt{\Delta}}{2} & \text{cuando } \Delta \geq 0, \\ \min(d_x, d_y) + W(\min(D)) & \Delta < 0 \end{cases} \quad (3)$$

En donde

$$d_x = \min(D(n_{right}, n_{left})) \quad (4)$$

$$d_y = \min(D(n_{up}, n_{down})) \quad (5)$$

y con

$$\Delta = (2 * W(\min(D))) - (d_x - d_y)^2 \quad (6)$$

Los valores contenidos en el vector de distancias  $D\{0, 1, \dots, r\}$  están definidos por la velocidad de expansión en el vector  $W\{0, 1, \dots, r\}$ . Si dichos valores son normalizados en  $D_{norm}\{0, 1, \dots, r\}$ , obtiene un valor proporcional a la normalización del tiempo de llegada de la onda a cualquier posición en el mapa.

En este sentido se generan tres tipos de valores según el tiempo requerido para alcanzar cualquier posición.

En el lugar en que se inicia la propagación del algoritmo FMM se tiene que  $t = 0$ . Dado que es la posición inicial, es alcanzada en el tiempo 0.

Luego se encuentran los valores comprendidos entre  $0 < t < 1$ , que son los tiempos necesarios para alcanzar cualquier posición en el mapa, en donde el valor más cercano a 1 corresponde a la posición más lejana respecto al punto de inicio del algoritmo.

Finalmente está el caso en que  $t = 1$ , este valor se reserva para las posiciones que se encuentran

<sup>1</sup> <http://es.mathworks.com/matlabcentral/fileexchange/6110-toolbox-fast-marching>



ocupadas o no pueden ser alcanzadas. En estos puntos la velocidad  $W$  es cero y por tanto, los tiempos de llegada tienden al infinito. Por esta razón son acotados y se les asigna el valor de 1.

Este proceso se realiza asegurando que se mantienen los protocolos de datos empleados en los mapas de ROS, en donde para  $D_{norm}(r) = 1$  el valor asignado para en el mapa de tiempos es  $T(r) = 100$  y para  $D_{norm}(r) < 1$  se calcula mediante la expresión

$$T(r) = 99 * \left( \frac{D(r)}{\max(D/D! = 1)} \right) \quad (7)$$

En su forma básica, el FMM recorre los píxeles de igual forma que se realiza en el algoritmo de Dijkstra, sin embargo, gracias a que las distancias entre píxeles son calculadas mediante la solución de la ecuación Eikonal, se logran eliminar efectos asociados a la fuerte discretización de los otros métodos. Las diferencias entre la forma de expansión y los efectos de discretización pueden ser estudiados en [7].

En la Figura 2 se muestra el mapa  $T\{0, 1, \dots, r\}$ , obtenido mediante el algoritmo FMM, en diferentes escenarios y con diferentes posiciones de origen. En la imagen, los píxeles azules indican posiciones temporalmente cercanas al punto de inicio, los píxeles rojos señalan aquellos que se encuentran más lejos, y por último, los píxeles amarillos muestran las posiciones inalcanzables. Es posible observar como en la figura inferior el escenario real tiene una variación menos uniforme de los tiempos en comparación a los mapas sintéticos de la figura superior, esto es causado por las zonas de baja densidad de píxeles en los extremos del mapa, en donde el costo de desplazamiento se incrementa y por tanto se disminuye en forma considerable la velocidad de movimiento en ese punto. En consecuencia, los tiempos para alcanzar dichos puntos, crecen notoriamente y alteran la normalización de  $T\{0, 1, \dots, r\}$ , causando que el resto del escenario se mantenga con un valor de tiempo más cercano al cero.

#### 4 Mapa de predicción de trayectorias

El segundo inconveniente del algoritmo Risk-RRT a tratar en este trabajo es la obtención de una representación probabilística para la predicción de trayectorias que tome en cuenta la disposición de los obstáculos en el mapa. En el trabajo original [1] se emplea un modelo de predicción basado en el procesamiento de la trayectorias mediante el filtro de Kalman [8]. Dado que en este cálculo solo se incluye la información de la trayectoria seguida por



Figura 2: Mapas de tiempo obtenidos mediante FMM.

el objetivo, la predicción no puede lidiar con cambios considerables de dirección. Por este motivo se hace inviable para realizar predicciones a largo plazo y/o en escenarios de complejidad elevada.

Con el fin de optimizar la predicción a largo plazo, se ha integrado un mapa de probabilidades que representa la probabilidad de desplazamiento del objetivo desde su posición instantánea hacia cualquier posición en el mapa. El mapa de probabilidades ha sido obtenido mediante una técnica de predicción de desplazamiento de peatones que a través de la generación de grafos es capaz de estimar la probabilidad para todas las rutas posibles que recorren el mapa desde la posición del objetivo. Este algoritmo es una extensión del algoritmo FMM y aprovecha las operaciones detalladas en la sección 3 para realizar las predicciones. Para profundizar en la descripción de este algoritmo es posible referirse a [9].

De este modo el componente probabilístico de la selección de nodos en el algoritmo Risk-RRT no solo está basado en la trayectoria actual del objetivo si no que toma en consideración la estructura morfológica del escenario.

## 5 Integración en Risk-RRT

Como se ha mencionado anteriormente, el algoritmo RRT asigna un peso para cada nodo del árbol (i.e. cada configuración del espacio de búsqueda). Cuando el árbol se completa o el tiempo para la planificación se termina, el nodo con el mayor peso se selecciona como la próxima meta y la trayectoria para alcanzarlo se extrae del árbol.

La integración de los algoritmos presentado se basa justamente en la asignación de pesos a las configuraciones del árbol, teniendo en cuenta la información obtenida a partir del algoritmo FMM y el mapa de probabilidades expuestos en secciones anteriores.

La primera modificación radica en que la distancia entre los nodos y el objetivo no se calcula mediante la distancia euclídea entre ambos. En su lugar, los nodos obtienen la distancia del mapa de distancias que resulta de emplear el algoritmo FMM descrito en la sección 3. Así las ventajas de baja discretización y la no existencia de mínimos locales son extendidas en la construcción del árbol.

El segundo punto de optimización del algoritmo busca extender sus capacidades de predicción. Con este fin se ha añadido el mapa de probabilidades de la sección 4 en el cálculo de los pesos del nodo. Añadir una componente de predicción a largo plazo en la selección de la ruta mejora las capacidades de respuesta del árbol en escenarios de alta complejidad y distancias elevadas. La mayor ventaja en este caso radica en que la predicción se realiza sobre todas las rutas posibles para alcanzarlo.

## 6 Experimentos y Resultados

Para demostrar la capacidad de integración de los algoritmos se ha fijado un escenario en el que un robot debe interceptar un objetivo señalado en el mapa. En la Figura 3 se muestra la planificación de trayectorias del algoritmo RRT con diferentes configuraciones de funcionamiento y la posiciones del robot y del objetivo señaladas con iconos de color azul y rojo respectivamente. El mayor reto de este experimento está representado por la configuración del escenario y la posición inicial del robot. Este se ha ubicado en un mínimo local y a su vez, la posición del objetivo se ha dejado estática para apreciar con más claridad los efectos de las diferentes etapas de integración de los algoritmos.

En la primera fila se muestran tres instantes de tiempo de la planificación realizada por el algoritmo original. Se puede observar que por sí solo

el algoritmo no es capaz de resolver fácilmente el problema de salir del mínimo local, dado que su nodo más probable está determinado por la distancia euclídea y no toma en cuenta la presencia de obstáculos. Para resolver este problema es posible modificar la parametrización del algoritmo Risk-RRT para generar un árbol con mayor distancia entre nodos o una apertura mayor del mismo. Sin embargo, cuando se aumenta la distancia mínima entre nodos se pierde resolución en la navegación del robot y se genera dificultades al maniobrar en pasillos pequeños. Por otro lado, cuando el problema se intenta resolver aumentando indiscriminadamente la extensión del árbol se genera un costo computacional muy elevado.

En la segunda fila se muestra la integración del algoritmo de RRT con el mapa  $T\{0, 1, \dots, r\}$  obtenido de FMM. Manteniendo la configuración inicial del árbol en el algoritmo original se puede apreciar como el cambio de la distancia euclídea a por una distancia de recorrido mejora sustancialmente el desempeño del algoritmo, llevándolo a una solución satisfactoria para alcanzar el objetivo. En este punto se encuentra también un crecimiento más rápido del árbol ocasionado por la supresión de los cálculos de distancia euclídea en la generación de cada nodo. Como se mencionó anteriormente, la existencia de un solo mínimo global en la expansión del mapa obtenido mediante FMM permite al robot salir fácilmente del mínimo local alcanzado con el algoritmo original y encontrar una trayectoria adecuada.

Finalmente, en la última fila se muestra el comportamiento obtenido en la expansión del árbol cuando se añade el peso del mapa de probabilidades. En las figuras, una probabilidad más alta se muestra como zonas de mayor claridad, en contraste con zonas de menor probabilidad con colores más oscuros. Se ha dado un pequeño incremento a la probabilidad de que el objetivo se desplace por la ruta de la izquierda, esto con el fin de clarificar el comportamiento del árbol en presencia de bifurcaciones. En las figuras se muestra como la expansión del árbol se realiza en su mayoría en la zona mayor probabilidad al alcance del robot. Esto permite al algoritmo concentrar los recursos computacionales en el descubrimiento de rutas en las zonas en que es más probable interceptar el objetivo en relación a las posibles rutas por las que este se puede desplazar.

Este es un primer paso en la implementación de técnicas basadas en FMM para la optimización de las tareas de interceptación de objetivos en la protección de infraestructuras críticas. El alcance de este trabajo está limitado a demostrar la factibilidad de la optimización de la tarea de in-

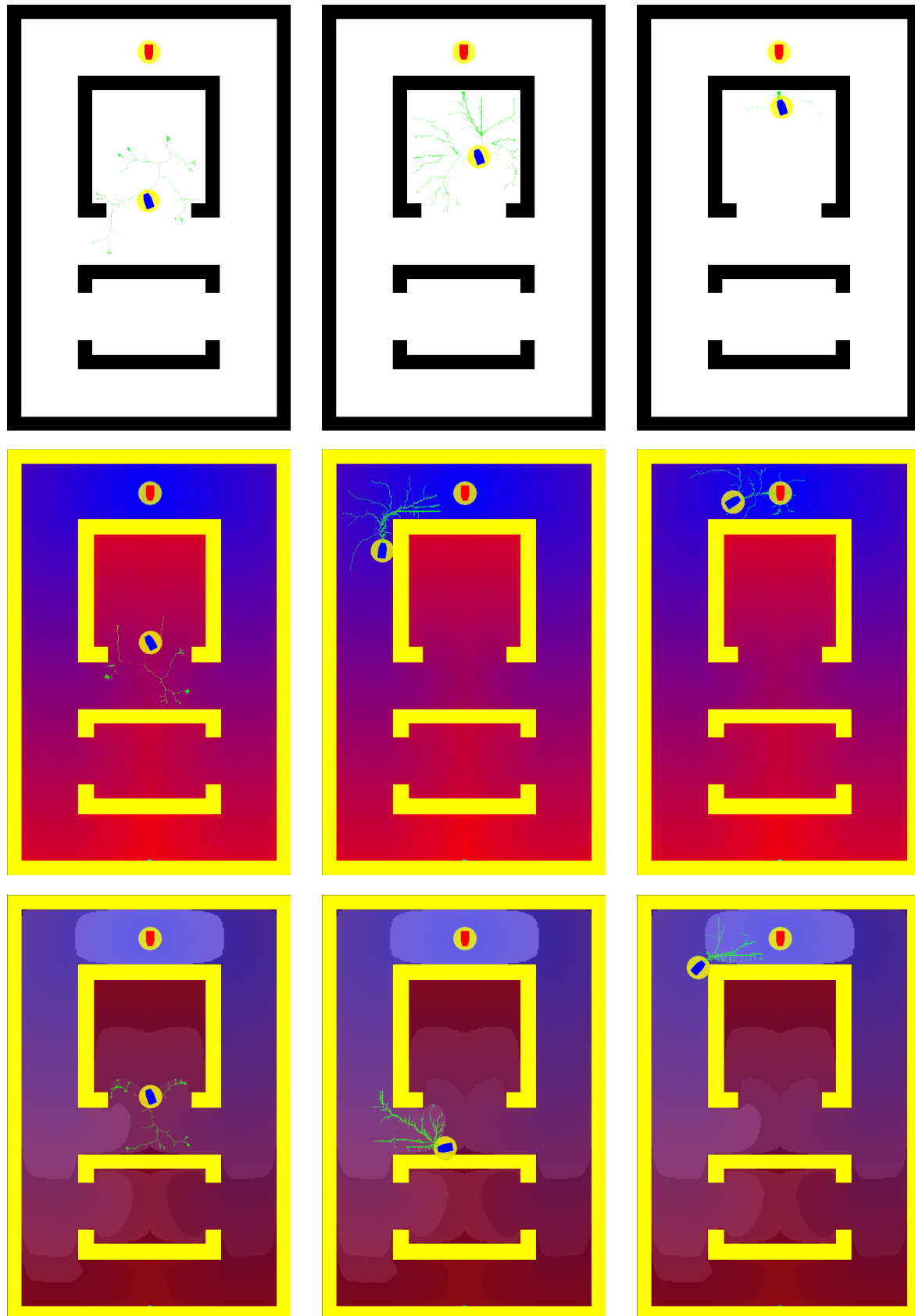


Figura 3: Trayectorias obtenidas con RRT, RRT+FMM y RRT+FMM+Predicción

terceptación realizada por el algoritmo Risk-RRT. Trabajos posteriores se dedicarán a la obtención de resultados cuantitativos sobre la integración propuesta.

## 7 Conclusiones

Se comprobó la compatibilidad y buena integración del algoritmo Risk-RRT, el Método de Fast Marching y la predicción de trayectorias basada también en FMM. Los algoritmos comparten la misma información de entrada y no requieren consideraciones adicionales para su funcionamiento conjunto.

Se ha demostrado como la inclusión intrínseca de la información aportada por el escenario mejora en gran medida el comportamiento del algoritmo Risk-RRT. Se encontró que la información aportada por el mapa en el algoritmo original es escasa y solo permite realizar las tareas de evasión de obstáculos. Al reemplazar dicho mapa con el mapa obtenido de FMM se logró incluir información relevante para resolver situaciones con mínimos locales sin tener que modificar la etapa de planificación. También, la inclusión adicional del mapa de probabilidades ha permitido mejorar el desempeño del algoritmo en situaciones de predicción de largo alcance.

Finalmente se han logrado demostrar la potencialidad del algoritmo Risk-RRT para desempeñar tareas de interceptación de objetivos. Las modificaciones realizadas permitirán desarrollar trabajos futuros que incluyan la integración de mapas con nuevos modelos probabilísticos e información de interés en la selección de la ruta óptima.

## Agradecimientos

Este trabajo fue parcialmente apoyado por el Grupo de Robótica y Cibernética de la Universidad Politécnica de Madrid (España), siendo financiado por los proyectos: PRIC (Protección robotizada de Infraestructuras Críticas; DPI2014-56985-R), patrocinado por el Ministerio español de Economía y Competitividad y RoboCity2030-III-CM (Robótica aplicada a la Mejora de la Calidad de Vida de los Ciudadanos fase III; S2013 / MIT-2748), financiado por Programas de Actividades de I + D en la Comunidad de Madrid y cofinanciado por Fondos Estructurales de la UE.

## Referencias

- [1] Garzón, M., Fotiadis, E.P., Barrientos, A., Spalanzani, A. (2014). RiskRRT-Based Planning For Interception of Moving Objects in Complex Environments. In ROBOT2013: First Iberian Robotics Conference. Advances in Intelligent Systems and Computing, vol 253. Springer, Cham.
- [2] Janson, L., Schmerling, E., Clark, A., & Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. The International journal of robotics research, 34(7), 883-921.
- [3] Starek, J., Schmerling, E., Janson, L., & Pavone, M. (2014). Bidirectional fast marching trees: An optimal sampling-based algorithm for bidirectional motion planning. In Workshop on Algorithmic Foundations of Robotics.
- [4] Fulgenzi, C., Spalanzani, A., Laugier, C., & Tay, C. (2010). Risk based motion planning and navigation in uncertain dynamic environment.
- [5] LaValle, S. M., & Kuffner Jr, J. J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378-400.
- [6] Petres, C., Pailhas, Y., Petillot, Y., & Lane, D. (2005, June). Underwater path planning using fast marching algorithms. In Oceans 2005-Europe (Vol. 2, pp. 814-819). IEEE.
- [7] Peyré, G. (2011). The numerical tours of signal processing. Computing in Science & Engineering, 13(4), 94-97.
- [8] Garzón, M., Garzón-Ramos, D., Barrientos, A. & Cerro, J. (2016). Pedestrian Trajectory Prediction in Large Infrastructures - A Long-term Approach based on Path Planning. In Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics, 381-389.
- [9] Garzón Ramos, D. A., Garzón Oviedo, M., & Barrientos Cruz, A. (2016). Pedestrian Motion Prediction: A Graph Based Approach.
- [10] Sethian, James A. (1996). A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences, 93(4), 1591-1595.